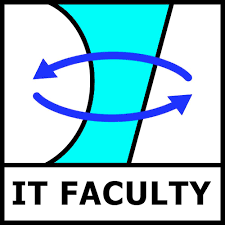
**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------



**BÁO CÁO ĐỒ ÁN  
CƠ SỞ NGÀNH MẠNG**

***ĐỀ TÀI***

1. **Giao tiếp các tiến trình bằng đường ống PIPE**
2. **Tìm hiểu và sử dụng kỹ thuật lập trình TCP và**

**xây dựng chương trình xem điểm thi theo mô hình Client - Server.**



**GVHD**: **TS. Phạm Minh Tuấn**

**SVTH** : **Trần Hữu Trung**

**MSSV** : **102150142**

***Đà Nẵng****, tháng 12/ 2018*

MỞ ĐẦU

Đồ án cơ sở ngành mạng là một trong những đồ án quan trọng của ngành Công Nghệ Thông Tin. Thông qua đồ án này, chúng em – là những sinh viên của trường đại học Bách Khoa Đà Nẵng sẽ đạt được những kiến thức nền tảng và có cơ hội được nghiên cứu chuyên sâu hơn về bộ môn *Lập trình mạng và Nguyên lí hệ điều hành.*

Trong đồ án này, về phần *Nguyên lí hệ điều hành*, em chọn đề tài nghiên cứu và tìm hiểu về cách thức giao tiếp giữa các tiến trình bằng cơ chế đường ống PIPE, còn phần *Lập trình mạng*, đề tài được chọn là tìm hiểu về kỹ thuật lập trình TCP và xây d ng chương trình xem điểm thi theo mô hình Client - Server. Với sự đồng hành và hỗ trợ nhiệt tình của các thầy cô bộ môn ngành Mạng, cùng sự nỗ lực cố gắng của bản thân, bản báo cáo *Đồ án cơ sở ngành mạng* nã hoàn thành đúng với thời hạn được giao.

Đặc biệt em xin trân trọng cảm ơn thầy Phạm Minh Tuấn đã hướng dẫn và chỉ bảo tận tình trong quá trình tìm hiểu, nghiên cứu và thực hiện đồ án này.

Em xin chân thành cảm ơn!

***Sinh viên thực hiện***

Trần Hữu Trung

MỤC LỤC

[**MỞ ĐẦU** 1](#_Toc501995693)

[**PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH** 3](#_Toc501995694)

[CHƯƠNG 1. CỞ SỞ LÍ THUYẾT 4](#_Toc501995695)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ 5](#_Toc501995696)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ](#_Toc501995697) 9

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 11](#_Toc501995698)1

[**PHẦN II: LẬP TRÌNH MẠNG** 12](#_Toc501995699)2

[CHƯƠNG 1. CƠ SỞ LÝ THUYẾT 12](#_Toc501995700)2

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 15](#_Toc501995701)5

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 17](#_Toc501995702)7

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 17](#_Toc501995703)9

[**PHỤ LỤC** 20](#_Toc501995704)

[**KẾT LUẬN CHUNG** 3](#_Toc501995705)1

# PHẦN I: NGUYÊN LÝ HỆ ĐIỀU HÀNH

**TIÊU ĐỀ: Giao tiếp giữa các tiến trình bằng cơ chế đường ống pipe**

Yêu cầu đề tài:

1. Giới thiệu tiến trình trong Linux, các hoạt động của tiến trình, cấu trúc tiến trình, giao tiếp giữa các tiến trình, các cơ chế giao tiếp và giới thiệu cơ chế liên lạc bằng đường ống Pipe.
2. Tạo đường ống giao tiếp (có thể sử dụng hàm Fork() ), các loại pipe.
3. Viết chương trình gồm 2 quá trình. Quá trình thứ nhất cho người dùng nhập vào từ bàn phím một chuỗi biểu diễn các phép tính gồm các phần tử +,-,(,). độ ưu tiên của các phép tính trong ngoặc, cặp dấu “(“ và “)” là cao nhất, phép +,và – có cùng độ ưu tiên. Ví dụ: 2\*9+(2\*3-4)-((3+7)/5)

Sau đó truyền chuỗi dữ liệu này sang quá trình thứ hai. Quá trình thứ hai thực hiện tính toán và trả về cho quá trình thứ nhất để thể hiện cho người sử dụng biết.

## **CHƯƠNG 1. CỞ SỞ LÝ THUYẾT**

* 1. **TIẾN TRÌNH TRONG LINUX**
* Tiến trình là một thực thể điều khiển đoạn mã lệnh có riêng một không gian địa chỉ, có ngăn xếp stack riêng rẽ, có bảng chứa các thông số mô tả file được mở cùng tiến trình và đặc biệt có một định danh PID (Process Identify) duy nhất trong toàn bộ hệ thống vào thời điểm tiến trình đang chạy.
* Để hoàn thành tác vụ của mình, một tiến trình có thể cần đến một số tài nguyên – như CPU, bộ nhớ chính, các tập tin và thiết bị nhập/xuất.Các tài nguyên này được cung cấp khi tiến trình được tạo hay trong quá trình thi hành.
* Các trạng thái của tiến trình
  + Trạng thái của tiến trình tại một thời điểm được xác định bởi hoạt động hiện thời của tiến trình tại thời điểm đó. Trong quá trình sống, một tiến trình thay đổi trạng thái do nhiều nguyên nhân như : phải chờ một sự kiện nào đó xảy ra, hay đợi một thao tác nhập/xuất hoàn tất, buộc phải dừng hoạt động do đã hết thời gian xử lý …
  + Tại một thời điểm, một tiến trình có thể nhận trong một các trạng thái sau đây :
    - **Mới tạo**: tiến trình đang được tạo lập.
    - **Running** : các chỉ thị của tiến trình đang được xử lý.
    - **Blocked** : tiến trình chờ được cấp phát một tài nguyên, hay chờ mộtsự kiện xảy ra.
    - **Ready**: tiến trình chờ được cấp phát CPU để xử lý.
    - **Kết thúc:** tiến trình hoàn tất xử lý.
* Cấu trúc của tiến trình

Hệ điều hành quản lí các tiến trình trong hệ thống thông qua khối quản lí tiến trình( Process Control Block – PCB). PCB là một vùng nhớ lưu trữ các thông tin mô tả cho tiến trình, với cá thành phần chủ yếu bao gồm:

* + Định danh của tiến trình: giúp phân biệt các tiến trình.
  + Trạng thái tiến trình; xác định hoạt động hiện hành của tiến trình.
  + Ngữ cảnh của tiến trình: mô tả các tài nguyên tiến trình đang trong quá trình, hoặc để phục vụ cho hoạt động hiện tại, hoặc để làm cơ sở phục hồi hoạt động cho tiến trình, bao gồm các thông tin về:
    - Trạng thái CPU: bao gồm nội dung các thanh ghi, quan trọng nhất là con trỏ lệnh IP lưu trữ địa chỉ câu lệnh kế tiếp tiến trình sẽ xử lý
    - Bộ xử lý: dùng cho máy có cấu hình nhiều CPU, xác định số hiệu CPU mà tiến trình đang sử dụng.
    - Bộ nhớ chính: danh sách các khối nhớ được cấp cho tiến trình.
    - Tài nguyên sử dụng: danh sách các tài mguyên hệ thống mà tiến trình đang sử dụng.
    - Tài nguyên tạo lập: danh sách các tài nguyên được tiến trình tạo lập.
  + Thông tin giao tiếp: phản ánh các thông tin về quan hệ của tiến trình với các tiến trình khác trong hệ thống :
    - Tiến trình cha: tiến trình tạo lập tiến trình này .
    - Tiến trình con: các tiến trình do tiến trình này tạo lập.
    - Độ ưu tiên : giúp bộ điều phối có thông tin để lựa chọn tiến trình được cấp CPU.
  + **Thông tin thống kê:** đây là những thông tin thống kê về hoạt động của tiến trình, như thời gian đã sử dụng CPU,thời gian chờ. Các thông tin này có thể có ích cho công việc đánh giá tình hình hệ thống và dự đoán các tình huống tương lai.
* Các thao tác điều khiển tiến trình

Hệ điều hành cung cấp các thao tác chủ yếu sau đây trên một tiến trình :

* + Tạo lập tiến trình bằng hàm system()
  + Thay thế tiến trình hiện hành với hàm exec()
  + Nhân bản tiến trình với hàm fork()
  + Kiểm soát và đợi tiến trình con
  + Đón xử lý tín hiệu khi tiến trình con kết thúc
  + Bỏ rơi tiến trình con
  1. **CƠ CHẾ GIAO TIẾP CÁC TIẾN TRÌNH**

Một tiến trình là một thực thể hoạt động của hệ điều hành thực thi các chương trình. Trong thực tế, có những tiến trình phức tạp và cần phải nhiều tiến trình giao tiếp để hoàn thành một công việc. Để làm việc cùng nhau, các tiến trình cần phải trao đổi dữ liệu. Vì vậy, cần phải có nhiều cơ chế giao tiếp interprocess (IPC) khác nhau. Một trong những cơ chế IPC cơ bản nhất là đường ống, tượng trưng cho dữ liệu tuần tự giữa các quá trình trong một đường ống.

* 1. **CƠ CHẾ GIAO TIẾP BẰNG ĐƯỜNG ỐNG – PIPE**

Các tiến trình chạy độc lập có thể chia sẻ hoặc chuyển dữ liệu cho nhau xử lý thông qua cơ chế đường ống (pipe):

* + Một pipe là một thiêt bị truyền thông tuần tự, dữ liệu có thể đọc (read) từ pipe cùng lúc ghi (write) lên pipe.
  + Pipe còn được dùng để liên lạc giữa hai thread trong một tiến trình hay giữa tiến trình cha và tiến trình con.
  + Trên đường ống dữ liệu chỉ có thể chuyển đi theo một chiều, dữ liệu vào đường ống tương đương với thao tác ghi (pipe write), lấy dữ liệu từ đường ống tương đương với thao tác đọc (pipe read). Dữ liệu được chuyển theo luồng (stream) theo cơ chế FIFO.
  + Một pipe là một kênh liên lạc trực tiếp giữa hai tiến trình: dữ liệu xuất của tiến trình này được chuyển đến làm dữ liệu nhập cho tiến trình kia dưới dạng một dòng các byte.
  + Khi một pipe được thiết lập giữa hai tiến trình, một trong chúng sẽ ghi dữ liệu vào pipe và tiến trình kia sẽ đọc dữ liệu từ pipe. Thứ tự dữ liệu truyền qua pipe được bảo toàn theo nguyên tắc FIFO.
  1. **ĐƯỜNG ỐNG GIAO TRIẾP TRAO ĐỔI DỮ LIỆU HAI CHIỀU**

Sử dụng cơ chế giao tiếp đường ống hai chiều dễ dàng cho cả hai phía tiến trình cha và tiến trình con. Trong cơ chế giao tiếp hai chiều bằng đường ống thì thường người ta sẽ dùng hai đường ống một chiều vì sẽ dễ xử lh hơn là sử dụng một đường ống. Khi sử dụng một đường ống phải tính toán sao cho tránh được tình huống nghẽn do cả hai phía cùng ghi dữ liệu vào đường ống. Trong đồ án này, nhóm chúng em sử dụng hai đường ống một chiều để thực hiện giao tiếp hai chiều giữa tiến trình cha và tiến trình con.

## **CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ**

Bài toán giao tiếp hai chiều giữa tiến trình cha và tiến trình con là một bài toán rất hay gặp và được ứng dụng rất nhiều. Tuy nhiên đối với dạng bài toán này nếu chỉ dùng một đường ống thì phải xử lý phức tạp, nguyên nhân là vì cơ chế đóng mở đầu khi đọc ghi của pipe. Chính vì vậy một giải pháp đơn giản cho bài toán này đó là sử dụng hai đường ống một chiều để giao tiếp.

Trong đó, một đường ống chỉ được dùng chuyên biệt cho việc nhận dữ liệu từ tiến trình cha, và đường ống còn lại được dùng để ghi ngược dữ liệu từ tiến trình con đến tiến trình cha.

**Bài toán:**

Viết chương trình gồm 2 quá trình. Quá trình thứ nhất cho người dùng nhập vào từ bàn phím một chuỗi biểu diễn các phép tính gồm các phần tử +,-,(,). độ ưu tiên của các phép tính trong ngoặc, cặp dấu “(“ và “)” là cao nhất, phép +,và – có cùng độ ưu tiên.

Ví dụ: 2\*9+(2\*3-4)-((3+7)/5)

Sau đó truyền chuỗi dữ liệu này sang quá trình thứ hai. Quá trình thứ hai thực hiện tính toán và trả về cho quá trình thứ nhất để thể hiện cho người sử dụng biết

**Tóm tắt mô tả bài toán:**

**Input:** Chuỗi phép tính str .

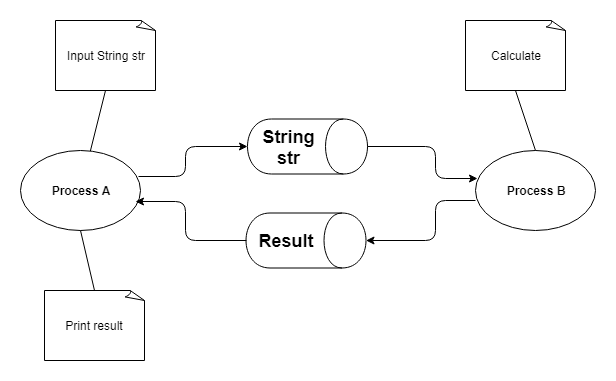
**Output:** Kết quả của phép tính vừa nhập vào.

Ví dụ bài toán: Tính giá trị của chuổi biểu thức 2\*9+(2\*3-4)-((3+7)/5) mà người dùng nhập vào thì:

Input: Chuối str =’ 2\*9+(2\*3-4)-((3+7)/5) ‘

Output: Chính là kết quả của phép tính trên, result = 18.

Từ yêu cầu bài toán như trên, em tổ chức thành hai tiến trình như sau:



*Hình 1 - Sơ đồ mô phỏng bài toán giao tiếp hai chiều giữa hai tiến trình bằng hai đường ống pipe.*

Các bước thực hiện của chương trình như sau:

* Bước 1: Tiến trình A sẽ nhận vào là chuỗi str sau đó thông qua đường ống gửi đến tiến trình B để kiểm tra.
* Bước 2: Tiến trình B sẽ nhận chuỗi str gửi đến từ tiến trình A và tiến hành xử lý tính toán. Sau đó tạo ra một báo cáo kết quả dạng chuỗi và thông qua một đường ống khác và gửi lại tiến trình A.
* Bước 3: Tiến trình A sẽ nhận chuỗi báo cáo và in kết qua ra màn hình.

Em sử dụng ngôn ngữ lập trình C trên hệ điều hành Linux để viết chương trình mô phỏng bài toán trên. Đối với ngôn ngữ C trên hệ điều hành Linux, để tạo ra một đường ống thì em sẽ làm như sau:

int filedes[2] ; // mảng chứa file descriptors ( filedes[0] : đọc, filedes[1]: ghi )

pipe(filedes); //tạo pipe

Với hàm pipe được cung cấp từ thư viện <**unistd.h**>. Kết quả trả về sẽ hai file descriptors của hai đầu đường ống. Nguyên tắc để giao tiếp qua đường ống là đầu đọc khi được mở ra thì sẽ chờ cho đến khi đầu còn lại ghi xong và đóng lại. Chính nhờ tín hiệu đóng của đầu kia mà đầu còn lại mới bắt đầu đọc dữ liệu.

Ngoài ra để tạo ra một tiến trình con trong ngôn ngữ C trên nền hệ điều hành Linux thì ta sử dụng hàm fork() để tạo như sau:

pid\_t new\_pid; // pid\_t: kiểu int

new\_pid = fork(); //tạo tiến trình mới

switch(new\_pid){

case -1:

//tien trinh khong duoc tao

break;

case 0:

//tien trinh con

break;

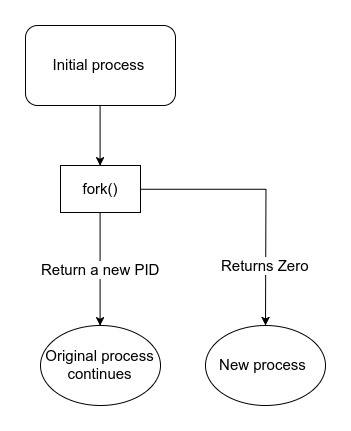
default:

//tien trinh cha

break;

}

Trong đó hàm fork() khi tạo mới thành công tiến trình con thì bên trong tiến trình con sẽ trả về trị 0, bên trong tiến trình cha sẽ trả về trị số nguyên là PID của tiến trình con vừa tạo, trong trường hợp không tách được tiến trình, hàm fork() sẽ trả về trị -1.



*Hình.2 - Sơ đồ mô phỏng quá trình tạo ra tiến trình con bằng hàm fork()*

Tiếp theo là quá trình trao đổi đọc ghi dữ liệu bằng đường ống giữa hai tiến trình cha và con. Đầu tiên chúng ta phải xác định đâu là đầu để đọc và đâu là đầu để ghi. Một quy tắc chung được dùng phổ biến là đầu 0 để đọc và đầu 1 để ghi.

Sau đó, trước khi ghi dữ liệu vào một đầu thì phải đóng đầu kia lại. Và tương tự, trước khi đọc dữ liệu từ đầu kia ra thì cũng phải đóng đầu đọc lại. Khi mở một đầu ra để ghi hay đọc cũng phải được xác định thông qua tham số “r” là đọc hoặc “w” là ghi.

**\* Thuật toán tính giá trị biểu thức:** ở đây em dùng stack chuyển biểu thức trung tố (infix) sang biểu thức hậu tố (postfix) rồi tính toán trên đó. Chi tiết thuật toán được trình bày ở mục 2 phần phụ lục. (\*)

**\* Code:** Từ cơ sở lý thuyết và các phân tích bằng mã lệnh C trên nền hệ điều hành Linux ở trên, em đã viết chương trình hoàn chỉnh. Mã lệnh đầy đủ của chương trình nằm ở mục 1 phần phụ lục.(\*)

## **CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ**

**Môi trường làm việc:**

- Hệ điều hành Ubuntu ( ở đây em dùng Bản Ubuntu 18.04)

- Ngôn ngữ lập trình C

- Công cụ IDE Sublime Text.

**3.1. CHẠY CHƯƠNG TRÌNH BÀI TOÁN GIAO TIẾP HAI CHIỀU GIỮA TIẾN TRÌNH CHA VÀ CON BẰNG HAI ĐƯỜNG ỐNG MỘT CHIỀU**

Từ mã lệnh ở phần phụ lục 1 trong báo cáo này, giả sử toàn bộ mã lệnh được lưu trong tệp tin **pipe.c**

* Trên hệ điều hành Linux chạy Terminal lên và dùng gcc để biên dịch chương trình:

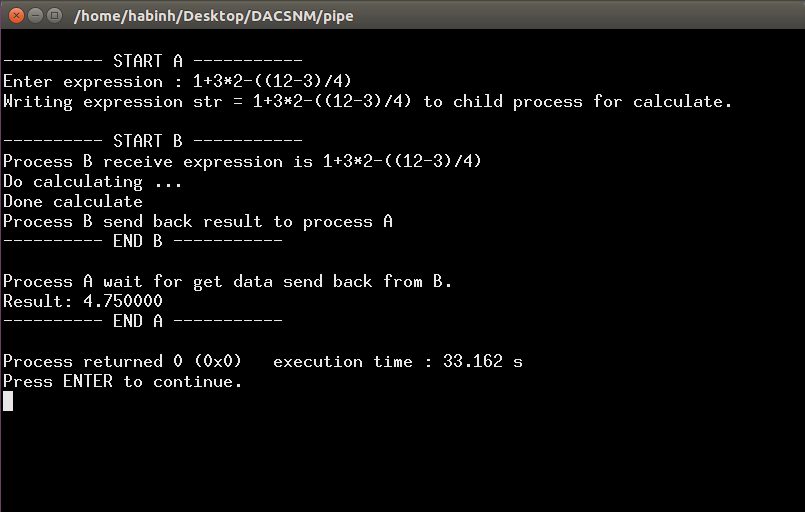
*gcc pipe.c -o pipe*

* Để thực thi chương trình thì chạy lệnh:

*./pipe*

* Người dùng nhập vào 1 chuỗi là biểu thức cần tính toán.
* Kết quả trả về cho người dùng và hiển thị trên terminal.

**Kết quả trả về sẽ như sau:**



*Hình 1.3: Kết quả demo chương trình pipe.*

**Nhận xét kết quả:**

* Bài toán trên với đầu vào biểu thức là chuỗi “2\*9+(2\*3-4)-((3+7)/5)”. Đoạn chương trình sử dụng hai đường ống một chiều để giao tiếp, chính vì vậy mà sau khi gửi dữ liệu xong, tiến trình A sẽ phải chờ cho đến khi dữ liệu được trả về từ tiến trình B để ghi ra màn hình thì mới kết thúc được.
* Kết quả đạt được chính xác với kết quả mong chờ của người dùng.

**3.2. ĐÁNH GIÁ**

**\* Ưu điểm:**

* Nắm được các kiến thức cơ bản về tiến trình trong linux và sử dụng được các hàm trong cơ chế giao tiếp các tiến trình bằng cơ chế đường ống pipe.
* Ứng dụng được để xây dựng chương trình giải quyết bài toán yêu cầu.

\* **Nhược điểm:**

* Chưa xây dựng được giao diện đẹp mắt và thân thiện với người dùng, chỉ dừng lại ở mức độ demo cho cơ chế giao tiếp giữa các tiến trình pipe.
* Exception ở quá trình tính toán chưa được xử lý thích hợp, chỉ mới thông báo lỗi nhập sai định dạng.

## **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

* **Kết luận:**

Bài toán ở trên vẫn còn khá đơn giản, trên thực tế còn có các kiểu giao tiếp như là giao tiếp giữa một chuỗi các tiến trình nối tiếp nhau, đồng bộ một tập tiến trình con vào một tiến trình đầu ra cũng rất hay gặp trong thực tế. Hoặc bài toán nối tiến trình phức tạp không theo dạng nối tiếp mà theo cơ chế bất đồng bộ.

Đặc biệt là vấn đề về thời gian chạy của CPU khi một tiến trình làm việc quá dài thì thời gian chuyển qua lại giữa các tiến trình có thể anh hưởng đến dữ liệu gây mất mát hoặc sai sót, ảnh hưởng tới kết quả mong muốn của chương trình.

* Những vấn đề đã đạt được:
* Trong quá trình làm đồ án, với những kiến thức đã học và tự tìm hiểu em đã hiểu hơn về các tiến trình và cơ chế giao tiếp giữa các tiến trình bằng đường ống PIPE.
* Mặt khác em đã cải thiện được kĩ năng lập trình với ngôn ngữ C và kĩ năng sử dụng Sublime Text.
* Những vấn đề chưa đạt được:
* Do thời gian có hạn và kiến thức của em còn hạn chế nên vẫn còn tồn tại một số vấn đề như giao diện chưa đẹp mắt, chưa test hết các exception cũng như bộ input lớn.
* **Hướng phát triển:**

- Tiếp tục nghiên cứu lý thuyết và thực hành.

- Khắc phục những vấn đề còn tồn tại chưa giải quyết được.

- Xây dựng thêm giao diện.

# PHẦN II: LẬP TRÌNH MẠNG

**TIÊU ĐỀ: Xây dựng chương trình xem điểm thi**

## **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**.

## **CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

* 1. **SƠ ĐỒ HOẠT ĐỘNG MÔ TẢ NGHIỆP VỤ HỆ THỐNG**
  2. **THIẾT KẾ**

## **CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ**

**Môi trường làm việc:**

* Hệ điều hành Ubuntu
* Java 8
* Công cụ IDE Netbean, xampp

**3.1. CHẠY CHƯƠNG TRÌNH XEM ĐIỂM THI**

**3.2. ĐÁNH GIÁ**

## **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

# PHỤ LỤC

**Mục 1. MÃ NGUỒN BÀI TOÁN GIAO TIẾP GIỮA CÁC TIẾN TRÌNH**

**Mục 2. THUẬT TOÁN TÍNH GIÁ TRỊ BIỂU THỨC**

# KẾT LUẬN CHUNG

Qua đồ án cơ sở ngành mạng này bọn em học và rút ra rất nhiều kiến thức bổ ích. Hiểu sâu hơn về nguyên lý vận hành và cách thức giao tiếp giữa các tiến trình, đặc biệt là cơ chế giao tiếp và kí thuật lập trình thông qua TCP/IP. Vận dụng được kiến thức học được vào việc tạo ra sản phẩm, thông qua đó kiến thức được củng cố tốt hơn. Về tiến độ công việc được giao, em cũng cơ bản hoàn thiện được đúng theo yêu cầu.

Em xin một lần nữa gửi lời cảm ơn chân thành đến thầy Phạm Minh Tuấn đã hướng dẫn tận tình để cho em hoàn thiện đồ án này.

*Xin chân thành cảm ơn!*

Trần Hữu Trung

**PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN :**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………